LOCK-IT-UP

Copy-Protection System

by

Jeff Gold

DOUBLE-GOLD SOFTWARE
13126 Anza Drive
Saratoga, CA 95070
408-257-2247

554-9133

Copyright (C) 1982 by DOUBLE-GOLD SOFTWARE

THIRD EDITION
FIRST PRINTING - 1982

All rights reserved. No part of this manual or the referenced software may be reproduced in whole or in part, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher.

Printed in the United States of America.

Lock-It-Up is a trademark of Double-Gold Software.

Apple is a registered trademark of Apple Computer Inc.

WARRANTY:

Double-Gold Software will replace any diskette contained within the Lock-It-Up System should it ever fail. Simply return the diskette to Double-Gold for fast replacement.

UPDATE POLICY:

Upon receipt of your product registration card, Double-Gold will automatically notify you of any future enhancements to this product. These enhancements will be made available to current owners for a nominal update fee.

NON-EXCLUSIVE LICENSE:

The purchase price of this product includes a non-exclusive license for the individual/firm purchasing this product to market their own programs using the protection features provided by this software and associated documentation.

Anyone desiring to use this program to protect software owned by individuals/firms other than the purchaser may do so only after receiving written consent from Double-Gold.

TABLE OF CONTENTS

CHAPTER 1: Introduction.................................... 1

CHAPTER 2: Getting Started................................. 3

CHAPTER 3: Initial Preparation............................. 7

CHAPTER 4: Master Creation................................. 9

CHAPTER 5: Diskette Duplication............................ 13

CHAPTER 6: Serial Number Check............................. 17

CHAPTER 7: Extra Tips...................................... 19

APPENDIX A: Program Listings............................... 21

CHAPTER 1
INTRODUCTION

Lock-It-Up is a sophisticated copy-protection and duplication system designed for the Apple II computer

The system requires an Apple II or II+ (or Apple III in emulation mode) with at least 48K bytes of RAM, at least two disk drives, and Applesoft in ROM or 16K card.

With the Lock-It-Up system you can make and duplicate as many different protected disks as needed.

The system is menu-driven resulting in a very simple process for copy-protecting a diskette.

The first step in creating a protected disk is to create a master. This master is created within the master creation subsystem. Within this subsystem you can vary several options in order to make a customized protection system for your diskette.

After a master of your program is created, Lock-It-Up offers a copy subsystem which is compatible with up to 14 disk drives. Within this subsystem you can specify complete verification which will guarantee that all of your data is correct. Also, sequential serial numbers can be applied to the disks during copy.

The third subsystem in Lock-It-Up is used for checking the serial numbers on diskettes protected with the system.

In addition, several tips are given throughout this manual which can be used to make your programs and data even more secure. To help further explain these techniques, a diskette is included that contains some sample Applesoft programs which use some of these techniques.

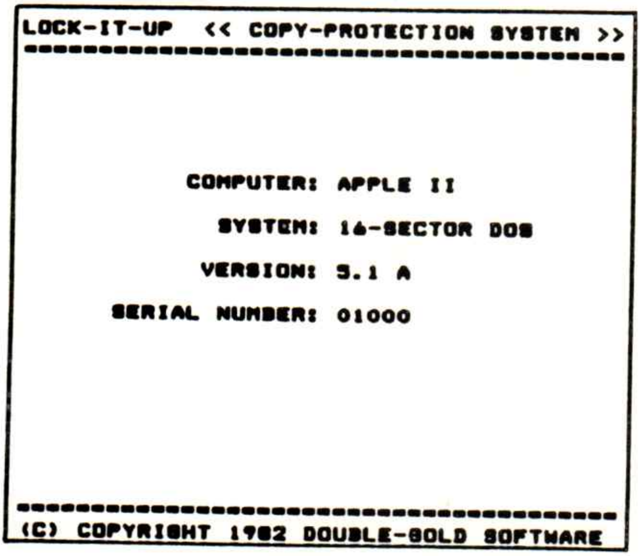
In addition to the above, several other helpful features are included with the Lock-It-Up system: a standard DOS 3.3 diskette can be used for data storage, a standard DOS 3.3 data diskette (without DOS) can be made automatically from your programs, and transfers, copying, etc. will be carried out at a greater speed. This is not compatible with all disk drives; therefore, if you choose it and you get numerous errors, we recommend that you do not use it. This option does not, however, affect the seeking speed of diskettes created with Lock-It-Up.

A menu of three options should now appear on the screen (figure 2.2). Simply choose the desired option by pressing the corresponding number.

NOTES ON NOTATION:

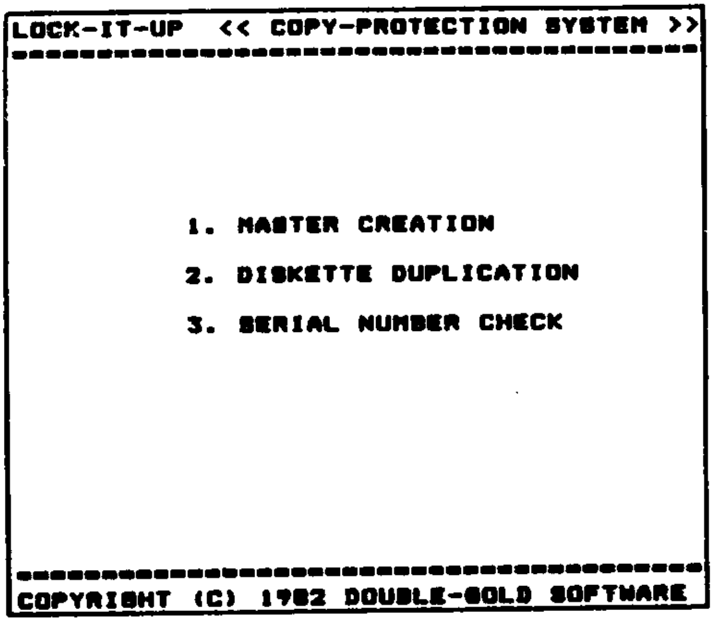
Default inputs are shown in inverse video. If you desire the default, you may simply press <RETURN>. Also, do not press the <RETURN> key after single-character inputs (except when pressing <RETURN> to choose a default). It is necessary, however, to press <RETURN> after entering an input line.

The majority of this manual assumes that your programs are written in basic. For machine language programmers, we have included extra information enclosed by (and). This information should not be typed into a basic program.



```
LOCK-IT-UP  << COPY-PROTECTION SYSTEM >>
========================================




            COMPUTER: APPLE II

              SYSTEM: 16-SECTOR DOS

             VERSION: 3.1 A

       SERIAL NUMBER: 01000




========================================
(C) COPYRIGHT 1982 DOUBLE-GOLD SOFTWARE
```

FIGURE 2.1



```
LOCK-IT-UP  << COPY-PROTECTION SYSTEM >>
----------------------------------------



           1. MASTER CREATION

           2. DISKETTE DUPLICATION

           3. SERIAL NUMBER CHECK






----------------------------------------
COPYRIGHT (C) 1982 DOUBLE-GOLD SOFTWARE
```

FIGURE 2.2

CHAPTER 3
INITIAL PREPARATION

Before a master can be created with the Lock-It-Up system, the following criteria must be met by your standard diskette:

1. Your diskette must be formatted under the normal DOS 3.3 operating system (before extra protection features are added, see chapter 6).

2. If your program will access a standard DOS 3.3 diskette from the protected diskette, you must add the following calls to your program in order to set the correct DOS. Their exact placement is not critical.

```
    before accessing a DOS 3.3 diskette: CALL 47741 ($BA7D)
before accessing the protected diskette: CALL 47721 ($BA69)
```

3. If your program needs to INIT a DOS 3.3 data diskette, you must first be sure that DOS 3.3 is set (see step 2 above), and then use the INIT command as usual including a dummy file name. This disk will not contain DOS or a boot file, therefore giving you more room for data storage.

4. If your program uses the RWTS subroutine directly, and you wish that the used sectors be transferred to the protected master, you must mark these sectors as used in the VTOC of your diskette.

5. If you EXEC a file while not within a program and will be using the AUTORUN APPLESOFT option (described in chapter 4), you must add the following pokes to your program before and after the EXEC:

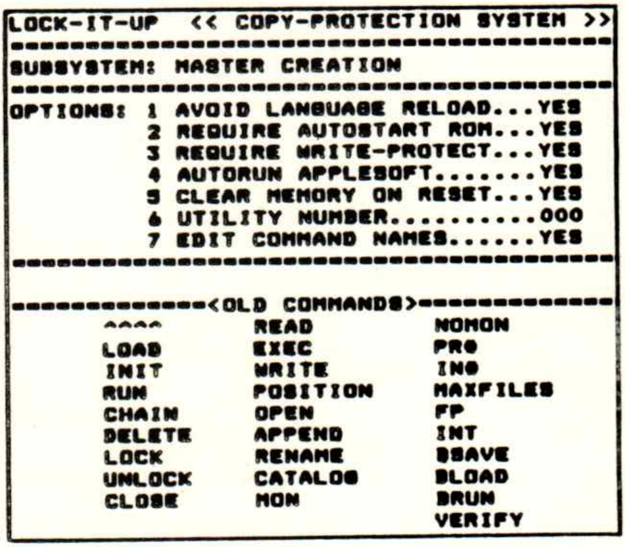
```
before exec: POKE 214,0       ($D6: $00)
 after exec: POKE 214,255     ($D6: $FF)
```

6. Your diskette cannot use the SAVE command. This has been permanently disabled; however, BSAVE can still be used.

7. You may optionally add any of the extra protection features described in chapter 7.

CHAPTER 4
MASTER CREATION

In order to create a protected master from your DOS 3.3 diskette, you must choose option 1 from the main menu. A new subsystem will then be loaded (figure 4.1). Follow the instructions below in order to create your protected master. Do not remove the Lock-It-Up diskette from the disk drive until you are prompted to insert your own diskettes.



```
LOCK-IT-UP  << COPY-PROTECTION SYSTEM >>
----------------------------------------
SUBSYSTEM: MASTER CREATION
----------------------------------------
OPTIONS: 1 AVOID LANGUAGE RELOAD...YES
         2 REQUIRE AUTOSTART ROM...YES
         3 REQUIRE WRITE-PROTECT...YES
         4 AUTORUN APPLESOFT.......YES
         5 CLEAR MEMORY ON RESET...YES
         6 UTILITY NUMBER..........000
         7 EDIT COMMAND NAMES......YES
----------------------------------------

-------------<OLD COMMANDS>-------------
      ^^^^       READ        NOMON
      LOAD       EXEC        PR#
      INIT       WRITE       IN#
      RUN        POSITION    MAXFILES
      CHAIN      OPEN        FP
      DELETE     APPEND      INT
      LOCK       RENAME      BSAVE
      UNLOCK     CATALOG     BLOAD
      CLOSE      MON         BRUN
                             VERIFY
```

FIGURE 4.1

1. After the master creation subsystem has been fully loaded, you have five YES/NO options to choose from. These are described in detail below. At this point, you may press <Y> for yes, <N> for no, <RETURN> for default, or <CTRL>-<C> to restart at each option.

OPTIONS:

(1) AVOID LANGUAGE RELOAD: If selected, this option will allow the language in a 16K card to remain active even after a re-boot. This option has no adverse effects if no 16K card is installed. You can use this option if your program requires a language which may not be resident in the users ROM. When using this option, you eliminate the necessity of having the necessary language resident on the diskette which you sell. The user can simply boot the "System Master" diskette, or equivalent, in order to load the necessary language into the 16K card, and then type "PR#n" in order to boot your diskette. "n" equals the slot in which the disk controller is installed.

(2) REQUIRE AUTOSTART ROM: If selected, your program will not be able to run on a computer that does not contain autostart ROM. All new Apple Computers are sold with the autostart ROM, however, some of the older computers may not be compatible with this requirement. This option allows a greater degree of protection, however, it may limit the number of people able to use a program which you are selling.

(3) REQUIRE WRITE-PROTECT: If selected, this option requires that the write-protect tab be in place on the protected diskette in order for the diskette to boot correctly. This option cannot be used if the user must be able to write data to the protected diskette. This option allows a high degree of protection when selected. Even if a person is able to make a perfect duplicate of a Lock-It-Up protected diskette (not currently possible), it is unlikely that they would put a write-protect tab on the diskette before testing it. This would result in the unauthorized copy not operating. You must remember to write-protect your diskettes before selling them if you use this option.

(4) AUTORUN APPLESOFT: If this option is selected, and a person is able to break out of your program and go to command level, whatever they type will cause the Applesoft program in memory to be executed again. It is recommended that you ALWAYS use this option, even if your program is in binary or integer basic, as it adds a high degree of protection.

(5) CLEAR MEMORY ON RESET: If selected, and the computer used has Autostart ROM and <RESET> is pressed, the memory of thr computer will be cleared and the currently logged slot will re-boot. This option will be cancelled if your program changes any of the reset vector points.

2. You will now be prompted for a utility number. This number can range in value from 0 to 255. If you hit <RETURN> the number will remain at 0. An explanation of this number's use is

given in chapter 7. If you are not using this option, the value of this number is unimportant.

3. You will now be prompted whether or not you want to edit command names. If you type "N" then go on to step 4.

EDIT COMMAND NAMES OPTIONS:

<- : moves cursor up one name
-> : moves cursor down one name
<C> : change command name (new names must be the same length as the old names)
<D> : delete toggle command (a row of ^'s denotes a deleted command)
<O> : display old command names
<N> : display new command names
<CTRL>-<X> : exit edit commands mode
<CTRL>-<C> : restart master creation system

If you will be using the Applesoft CHAIN program, you should not change or delete the LOAD command.

4. You will now be asked if everything is correct. If it is, hit <Y> or <RETURN> accepts it as a default. If you type <N> the master creation subsystem will restart.

5. The next question will be "FORMAT DEVELOPMENT DISKETTE?" which is defaulted to NO. A development diskette is a standard DOS 3.3 diskette which simply has the changed names and the two "calls" which select between normal and protected DOS set so that they will not generate errors (as in normal DOS 3.3). If you desire to use this option, continue reading at step 6 under the title "FORMATTING DEVELOPMENT DISKETTE ONLY" on the next page.

6. The computer will now display the message "<<LOADING FILES>>" as it reads some more data from the Lock-It-Up diskette.

7. You will now be prompted for a five character I.D. stamp. You may type any legal alpha-numeric characters for the I.D. You must remember this stamp because it is needed at the time you wish to duplicate more diskettes.

8. You will now be asked for the name of the "boot file" on your diskette. This is normally "HELLO". Type in the file name which you are using.

9. You will now be prompted for the file type. A boot file can be either an <A>pplesoft file, <I>nteger file, <B>inary file, or <T>ext file (exec). NOTE: The abbreviations used are different than those used in Lock-It-Up 5.0 and earlier versions.

10. The next prompt is whether you want to use fast sector interleaving. If you use this option, access (loading or saving) of basic and binary files will be much faster on the protected diskette. Text files, however, on the protected diskette will

work much more slowly. The access speed to a standard DOS 3.3 diskette is not affected by this option.

11. You will now be prompted for the source and destination slots and drives. Single drive transfer is not possible.

12. You will now be prompted to insert your source (DOS 3.3) and a blank diskette (to become the protected master). Hit a key to begin the transfer.

13. The screen will display, in order: <<COMPILING DATA>>, <<FORMATTING>>, <<READING>>, <<ENCRYPTING>> as it begins the transfer process. The screen will then display which tracks and sectors are being transferred. This process can take up to twenty minutes on a full diskette if fast seeking is not used.

14. After your protected master is completed, you may hit a key to reboot the disk drive.

FORMATTING DEVELOPMENT DISKETTE ONLY:

6. The screen will now display "<<LOADING FILES>>" as more data is read from the Lock-It-Up diskette.

7. You will next be prompted for the boot file and file type as in steps 8 and 9 above.

8. You will now be prompted for the slot and drive which you wish to format.

9. You will now be prompted to insert a blank diskette into that drive and hit a key to format that diskette.

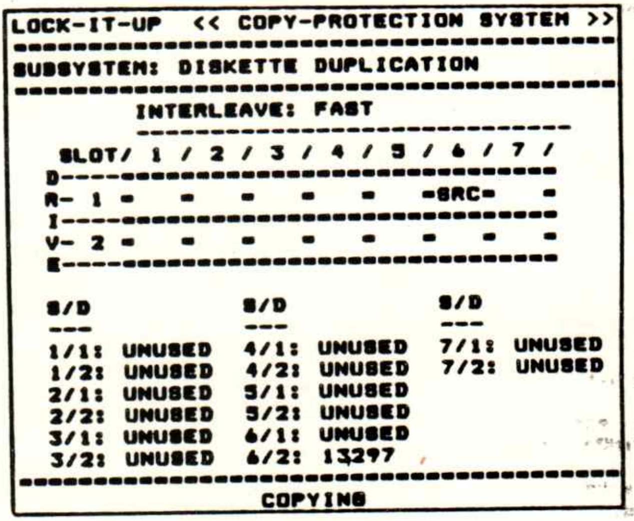
10. The screen will display <<COMPILING DATA>>, and then <<FORMATTING>> as the disk is created.

11. Hit a key to reboot the drive after the development diskette is created.

NOTE: Although you specify a boot file and type for the diskette, no file is stored on the diskette. You may use this development diskette for development with the changed names and then use it as a source for creating an actual master, if necessary.

CHAPTER 5
DISKETTE DUPLICATION

To use this option, you must select option 2 from the main menu. The duplication program will be loaded and the duplication screen (figure 5.1) will be displayed. Follow the steps listed below.



```
LOCK-IT-UP  << COPY-PROTECTION SYSTEM >>
-----------------------------------------
SUBSYSTEM: DISKETTE DUPLICATION
-----------------------------------------
         INTERLEAVE: FAST
         -----------------------------
   SLOT/ 1 / 2 / 3 / 4 / 5 / 6 / 7 /
   D------------------------------
   R- 1 -   -   -   -   -   -SRC-  -
   I------------------------------
   V- 2 -   -   -   -   -   -   -  -
   E------------------------------

   S/D           S/D           S/D
   ---           ---           ---
   1/1: UNUSED   4/1: UNUSED   7/1: UNUSED
   1/2: UNUSED   4/2: UNUSED   7/2: UNUSED
   2/1: UNUSED   5/1: UNUSED
   2/2: UNUSED   5/2: UNUSED
   3/1: UNUSED   6/1: UNUSED
   3/2: UNUSED   6/2: 13297
-----------------------------------------
                COPYING
```

FIGURE 5.1

1. The display on your screen is layed out to show all of the possible slots and drives which can possibly contain a disk drive. If you set a drive position as available, that particular position will be shown by a blank field. If a drive position is set as not available, that position will be shown in inverse field. The source drive which you designate will contain the letters "SRC". The display is defaulted to slot 6, drive 1 as the source and slot 6, drive 2 as the only destination drive. To change this set-up, use the following keys:

<- : move cursor left or up
-> : move cursor right or down
<Y> : specifies a used drive
<N> : specifies an unused drive
<S> : specifies the source drive
<A> : accepts the displayed configuration
<CTRL>-<R> : returns the default configuration to the screen

2. You will now be asked whether you want to back-up a duplication master. Since the master and the copies (which are the final products) are different, we have made this provision for backing-up your masters without going through the long process of master creation again.

3. You will now be asked for the five-character I.D. stamp which you assigned to the diskette during master creation.

4. Next you will be prompted for a starting serial number. The largest serial number that can be used is 99,999. Above that value the counter will be returned to 0.

5. You will now be asked whether you would like to have complete verification. This increases the time for copies, however, it also guarantees that the data on each copy is accurate and readable.

6. You will now be asked whether you used fast sector interleaving when you created your master.

7. Finally, you will be prompted to insert your diskettes. If you press <CTRL>-<R> at this point, you will be returned to step 2 above.

8. As the diskettes are being copied, the status of each drive is marked on the screen: <F>ormatting, <W>riting, <V>erifying, or a flashing "SRC" when reading.

9. If a read error occurs, the copy will be aborted and you will be returned to step 7 above.

10. If a write error occurs, the drive will be marked by a flashing "ERR" message and copying will continue to those drives which are not errored. The serial number of the drive which got the write error will be displayed in inverse video.

11. If a diskette fails to verify, the program will attempt to re-write the data. If that still fails, the program will treat that disk as if it had received a write error as in step 10.

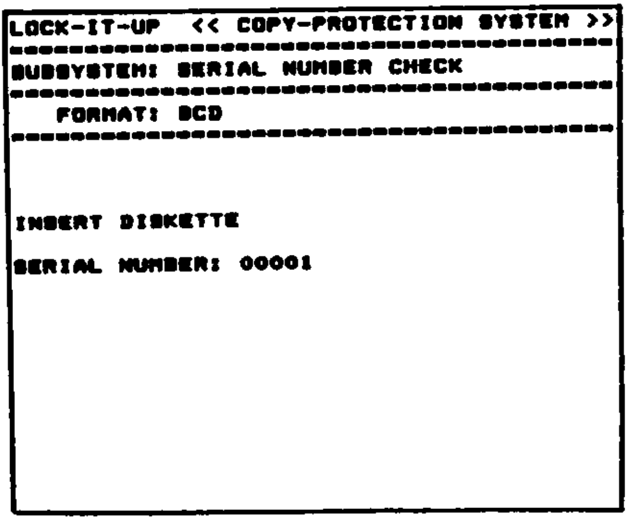
12. When a copy is completed, the serial numbers will be updated. Inverse serial numbers denote errored copies.

13. After a copy is complete, the computer will beep and prompt you to insert new diskettes. Typing <CTRL>-<R> will send you back to step 2 above. Any other key will continue the duplication process.

CHAPTER 6
SERIAL NUMBER CHECK

The last program supplied on the Lock-It-Up system diskette is used for checking serial numbers. Select option 3 from the main menu. A screen as in figure 6.1 will be displayed. You will be prompted for the format in which the number is stored, BCD or FP. For this version of Lock-It-Up choose <B>CD. The floating-point option was added to allow compatability with earlier Lock-It-Up systems. To check the serial number on a diskette, simply insert the diskette and press a key.

If you would like to check the serial number from your own programs, you must use the RWTS subroutine found in DOS. The serial number is stored in unpacked BCD in the five bytes starting with byte 96 decimal or $60 hex (counting from 0) on track 0, sector 5. If you are not familiar with RWTS, you may modify and use our SERIAL NUMBER CHECK program provided on our Lock-It-Up demo disk and appearing as listing 1 in appendix A.



```
LOCK-IT-UP  << COPY-PROTECTION SYSTEM >>
----------------------------------------
SUBSYSTEM: SERIAL NUMBER CHECK
----------------------------------------
   FORMAT: BCD
----------------------------------------



INSERT DISKETTE

SERIAL NUMBER: 00001
```

FIGURE 6.1

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

CHAPTER 7
EXTRA TIPS

1. Be sure to delete, or at least change as many DOS command names as possible (see chapter 4). For best protection, try not to use the commands: FP, INT, or BSAVE in your programs. If your programs require any of these commands, it would be best to change their names.

2. For added protection, set "DOS 3.3 mode" at the beginning of your program (CALL 47741) and switch to protected DOS only when necessary.

3. Add the following call to your program in at least one place. This call will attempt to INIT the currently logged diskette if your program is executed from a standard DOS 3.3 diskette (for example, if a person successfully translates the code to DOS 3.3 using special hardware). It will have no effect when executed from a protected diskette, no matter which DOS is selected.

```
CALL PEEK (40222) + PEEK (40223) * 256 + 1
```

4. Put DOS commands into REM statements so that they will be executed if the program is listed. For example, to put the "FP" command into the first line of your program, type the following after your program is loaded. Your program should not use line 0 when using this example.

```
]0 REM++FP

]CALL -151

*806:D 4

*3D0G

]SAVE name
```

NOTE: Do not type any spaces when entering line 0. "name" represents the name of your program.

5. Put machine language routines into REM statements.

6. If you would like to have the computer clear memory and reboot, use "CALL 46939" (JMP $875B).

7. If possible, make your program so that it does not reside entirely in memory at one time (see CHAIN in the DOS manual). If your program resides entirely in memory at one time, and does no more disk accessing, use the following commands in your program as soon as all disk accessing is completed.

```
PRINT CHR$(4)"BLOAD CMR" : CALL 904
```

The file CMR must reside on your protected diskette and may be transferred from our DEMO disk. This program resides from locations 904-968 ($388-$3C8) (see listing 2 in Appendix A). When it is called, it will clear DOS out of the computer, and modify itself so that it will act like the clear memory and reboot routine previously present in the cleared DOS. In order to clear memory and reboot from your own programs, you must now type CALL 904 ($388) a second time. Be sure NOT to use any calls into DOS that are given in this manual (set DOS mode, etc.), after calling this routine, or your program will not work.

8. You may append the program titled "PROTECT.APPEND LINES 0-2" to your program. It will occupy lines 0-2 of your program. These lines use the features described in steps 3-5 above. Do not attempt to run or list this program.

9. Make extensive use of ON ERR GOTO statements in your programs in order to intercept attempts by the user to stop the program by pressing <CTRL>-<C> or purposely causing other errors. This command is described in the Applesoft Basic Reference Manual.

10. Access locations in memory which are cleared when the <RESET> key is pushed. For example, text page locations and certain zero-page locations including byte 60 ($3C).

11. The utility number (see chapter 4) may be read in by your boot file by checking the contents of location 512 ($200). This can be done using the PEEK function of basic, as shown below. This must be done BEFORE the user is prompted for any type of input from the keyboard as any input will change the utility number.

```
UN = PEEK (512)
```

To make best use of this statement, use the utility number to assign a value to a variable in the range 0-255 that has a critical usage in your program. This is best achieved if the PEEK statement occurs as part of a complex formula.

APPENDIX A
PROGRAM LISTINGS

Listing 1

```
10  REM  ********************
15  REM  *                  *
20  REM  * A SAMPLE PROGRAM *
30  REM  *TO READ THE SERIAL*
40  REM  *  NUMBER FROM A   *
50  REM  *DISKETTE PROTECTED*
60  REM  *  WITH LOCK-IT-UP *
70  REM  *                  *
80  REM  ********************

82  REM
SET HIMEM TO PROTECT VARIABLES

84  HIMEM: 16383
90  REM
MACHINE LANGUAGE DATA

100  DATA  32,227,3,32,217,3
110  DATA  176,6,169,0,141,0
120  DATA  3,96,169,1,208,248
130  REM
PREDEFINED VARIABLES

139 VOLUM = 0
140 TRACK = 0
141 SECTR = 5
142 LOBUF = 0
143 HIBUF = 64
144 COMND = 1
150  REM
POKE DATA INTO MEMORY

160  FOR LOC = 0 TO 17
170  READ VAR
180  POKE 769 + LOC,VAR
190  NEXT LOC
```

continued on next page

Listing 1 (cont'd)

```
200  REM
SET RWTS PARAMETER TABLE

210  POKE 47083,VOLUM
220  POKE 47084,TRACK
230  POKE 47085,SECTR
240  POKE 47086,LOBUF
250  POKE 47089,HIBUF
260  POKE 47092,COMND
270  REM
READ SERIAL NUMBER

280  CALL 769
290  REM
CHECK FOR READ ERROR

300  IF  NOT  PEEK (768) THEN  GOTO 330
310  PRINT "READ ERROR  ": END
320  REM
PRINT SERIAL NUMBER

330  PRINT "SERIAL NUMBER = ";
340  FOR DIGIT = 0 TO 4
350  PRINT  PEEK (16480 + I);
360  NEXT DIGIT
370  END
```

Listing 2

```
**********************************
*                                *
*     CLEAR DOS MEMORY           *
*         ROUTINE                *
*                                *
* INVOKED BY:      CALL 904      *
*                                *
**********************************
*
               ORG  $388        ;START DEFAULT CODE AT $300
CNTR           =    6           ;SET PAGE COUNTER AT ZERO-PAGE LOCS 0-1
STRTHI         =    $9C         ;START MEMORY CLEAR  AT PAGE STRTHI+1
ENDHI          =    $BF         ;END MEMORY CLEAR AT PAGE ENDHI
BYTE           =    $03         ;USE THIS BYTE FOR VALUE TO STORE IN MEMORY
RSTSTHI        =    $03         ;USE THIS BYTE FOR BOTTOM OF CLEAR ON (RESET)
CODE           LDA  #$00
               STA  CNTR        ;INITIALIZE LOW BYTE OF COUNTER
               LDA  #ENDHI
               STA  CNTR+1      ;INITIALIZE HIGH BYTE OF COUNTER
               LDY  #00         ;USE Y REGISTER FOR LOW BYTE COUNTER
CONTCLR2       LDA  #BYTE       ;SET ACCUMULATOR TO VALUE FOR MEMORY CLEAR
CONTCLR        STA  (CNTR),Y    ;CLEAR BYTE OF MEMORY
               DEY              ;DECREMENT LOW BYTE COUNT
               BNE  CONTCLR     ;NOT DONE, CONTINUE CLEARING
               DEC  CNTR+1      ;DECREMENT PAGE COUNTER
MOD1           LDA  #STRTHI     ;GET HIGH BYTE OF PAGE TO END
               CMP  CNTR+1      ;IS IT DONE?
               BNE  CONTCLR2    ;NO, KEEP ON CLEARING
MOD2           JSR  $FE89       ;MAKE SURE TO SET I/O HOOKS TO NORMAL
               JSR  $FE93       ;MONITOR ROUTINES, NOT DOS.
               LDA  #RSTSTHI
               STA  MOD1+1      ;MODIFY CODE TO CLEAR ENTIRE COMPUTER
               LDA  #<CODE
               STA  $3F2
               LDA  #>CODE
               STA  $3F3
               JSR  $FB6F
               LDA  #$4C        ;SET ROUTINE ON (RESET) TO REBOOT
               STA  MOD2
               LDA  #$00
               STA  MOD2+1
               LDA  #$C6
               STA  MOD2+2
               RTS              ;RETURN TO CALLER
```

```
0388- A9 00 85 06 A9 BF 85 07
0390- A0 00 A9 03 91 06 88 D0
0398- FB C6 07 A9 9C C5 07 D0
03A0- F1 20 89 FE 20 93 FE A9
03A8- 03 8D 9C 03 A9 88 8D F2
03B0- 03 A9 03 8D F3 03 20 6F
03B8- FB A9 4C 8D A1 03 A9 00
03C0- 8D A2 03 A9 C6 8D A3 03
03C8- 60 34 35 36 37 38 40 39
```